

Blatt 5 - Gruppe: G1-07

Mike Lenz, Jonas Tesfamariam

4. Juni 2023

Aufgabe 1

```
select v.titel, count(s.matrnr) as anz_hoerer
from studenten s join hoeren h
on s.matrnr = h.matrnr join vorlesungen v
on h.vorlnr = v.vorlnr
group by v.titel
order by anz_hoerer desc limit 5;
```

```
select s.matrnr
from studenten s join pruefen p on s.matrnr = p.matrnr
where p.note < 5 group by s.matrnr order by avg(p.note);
```

```
select p.name
from professoren p join vorlesungen v
on p.persnr = v.gelesen_von
where v.titel
like '%systeme';
```

```
drop table assistenten;
```

```
alter table professoren
add gebaeude integer check (gebaeude > 0),
add raumnr integer check (raumnr > 0);
```

```
alter table pruefen
add ects integer;
```

```
update pruefen set ects = 2 * vorlesungen.sws
```

```
from vorlesungen  
where pruefen.vorlnr = vorlesungen.vorlnr;
```

Aufgabe 2

```
select gebaeude
from professoren p join pruefen pr on p.persnr = pr.persnr
group by gebaeude
order by count(*) asc
limit 1;
```

```
select distinct s.name
from studenten s join pruefen p on s.matnr = p.matnr
join voraussetzen v on p.vorlnr = v.nachfolger
left outer join pruefen p2
on v.vorgaenger = p2.vorlnr and s.matnr = p2.matnr
where p2.note is null;
```

```
select distinct s.name
from studenten s join pruefen p1 on s.matnr = p1.matnr
join pruefen p2 on s.matnr = p2.matnr and p1.vorlnr = p2.vorlnr
where p1.note > 4 and p2.note < 5 and p2.note >= (
    select min(note)
    from pruefen p
    where p.matnr = s.matnr and note < 5
);
```

```
(select name, 'Doz.' as vermerk
from professoren)
union
(select name, case
when (select count(*)
      from studenten s2 join pruefen p
      on s2.matnr = p.matnr
      where s1.matnr = s2.matnr and p.note < 5)
>= (select count(*)
     from vorlesungen)
then 'Abs.'
else null
end as vermerk
from studenten s1 );
```

Aufgabe 3

Wir verwenden folgendes Schema für 3b:

```
create table x(  
    a integer ,  
    b integer  
);  
create table y(  
    b integer ,  
    c integer  
);  
create table x_default(  
    a_default integer ,  
    b_default integer  
);  
create table y_default(  
    b_default integer ,  
    c_default integer  
);  
insert into x (a,b) values (1,2),(9,8);  
insert into y (b,c) values (2,3),(7,6);  
insert into x_default (a_default,b_default) values (0,0);  
insert into y_default (b_default,c_default) values (0,0);
```

Queries:

```
create table bool_table(  
    a boolean ,  
    b boolean  
);  
insert into bool_table (a, b) values (true, true),  
                                     (true, false),  
                                     (true, null),  
                                     (false, true),  
                                     (false, false),  
                                     (false, null),  
                                     (null, true),  
                                     (null, false),  
                                     (null, null);  
  
select a,b,a and b from bool_table;  
  
select
```

```
        coalesce(x.a, xd.a_default),  
        coalesce(x.b, y.b, xd.b_default),  
        coalesce(y.c, yd.c_default)  
from x_default xd, y_default yd, x full outer join y  
on x.b = y.b;
```

Aufgabe 4

```
insert into myview1(id, title, submission_date)
values (2, 'test', '2022-02-02');
```

```
insert into
  myview3 (id, title, submission_date, posting_id, no, content)
values (4, 'test2', '2023-04-04', 4, 1, 'this is content');
```

Bei myview2 ist es nicht möglich über die view Daten hinzuzufügen, da kein Titel angegeben werden kann und dieser bei posting *not null* ist.

Bei myview3 ist es möglich über die view Daten einzufügen, jedoch nur laut dem SQL:1999 Standard.

Bei myview4 ist es nicht möglich über die view Daten hinzuzufügen, da eine Gruppierung verwendet wird.